SmartHome

Progetto di Ingegneria del Software

Alessandra Rota 829775

Beatrice Stropeni 830017

Milano, Gennaio 2020

Indice

1 Specifiche

- il sistema considera e gestisce i seguenti oggetti:
  - lampade
  - finestre
  - tapparelle
  - oggetti per la pulizia:
    - lavatrice
    - lavastoviglie
    - robot
  - sensori:
    - movimento
    - fughe gas
    - intrusione
    - termostato
- il sistema, quando rileva fughe di gas, apre le finestre ed effettua una chiamata di emergenza
- il sistema, quando rileva intrusioni, effettua una chiamata di emergenza
- il sistema, quando rileva movimenti interni alla casa, effettua una chiamata di emergenza
- il sistema rileva la temperatura interna e, dopo averla valutata, decide se modificarla oppure no, riscaldando o raffreddando la stanza. Nel caso in cui il sistema debba riscaldare la stanza, ma le finestre sono aperte, la temperatura reagisce in base al numero di finestre aperte. Se ci sono finestre aperte, non possiamo chiuderle se viene rilevata una fuga di gas
- l'utente può:
  - accendere e spegnere l'allarme, le luci, il robot della pulizia, la lavatrice, la lavastoviglie
  - abbassare e alzare le tapparelle
  - aprire e chiudere le finestre
  - impostare la temperatura desiderata
- l'utente può programmare:
  - l'accensione della lavatrice, della lavastoviglie e del robot della pulizia in un determinato momento
  - la modifica della temperatura in un determinato momento
- le scelte semiautomatiche dell'utente hanno la priorità sui programmi da lui impostati
- ad eccezione di emergenze, le scelte semiautomatiche dell'utente hanno la priorità sulle scelte automatiche del sistema

2 Analisi

2.1 Diagramma dei Casi d'Uso

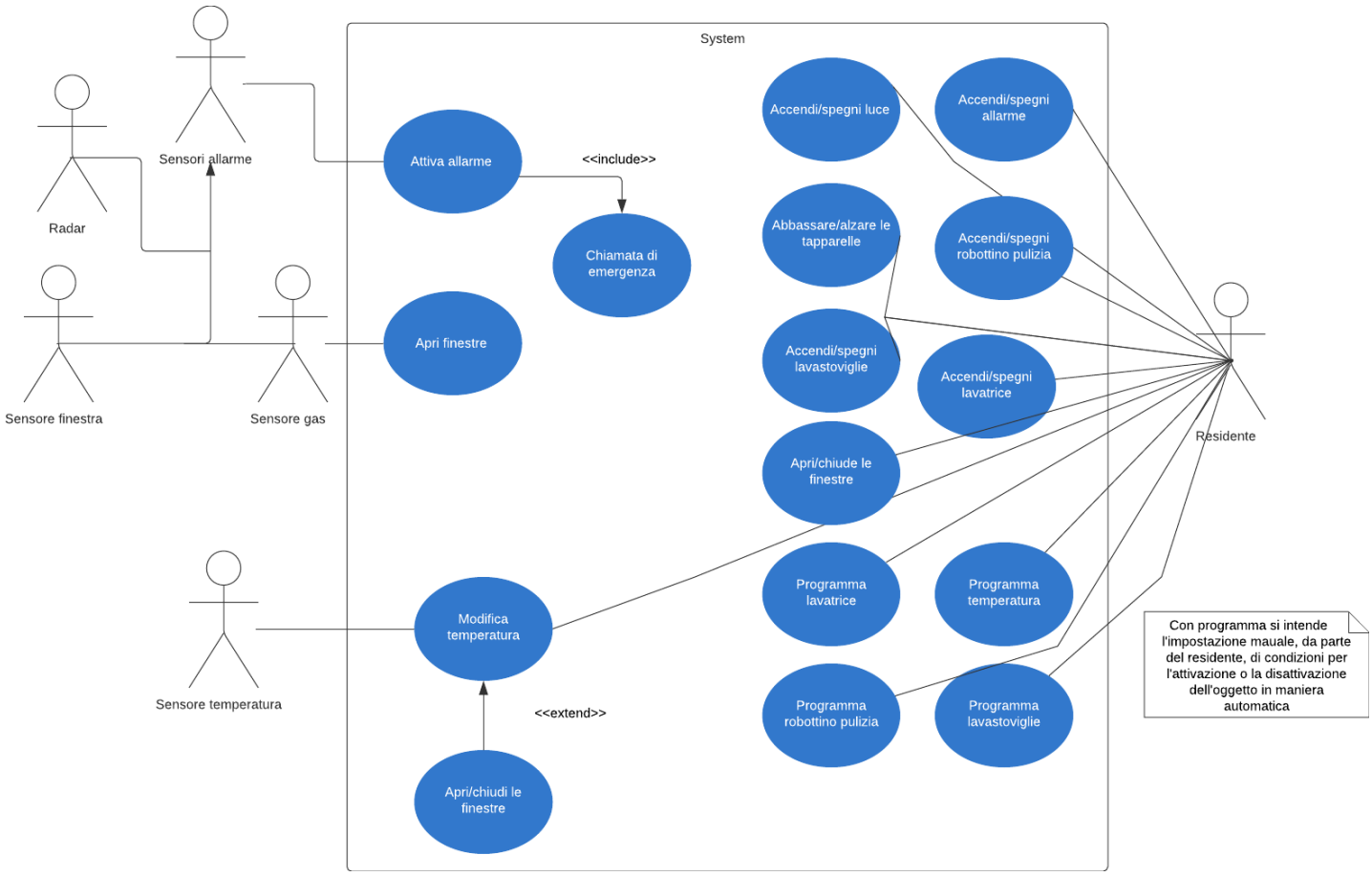


Figura 1: Diagramma dei Casi d'Uso

2.2 Modello di Dominio

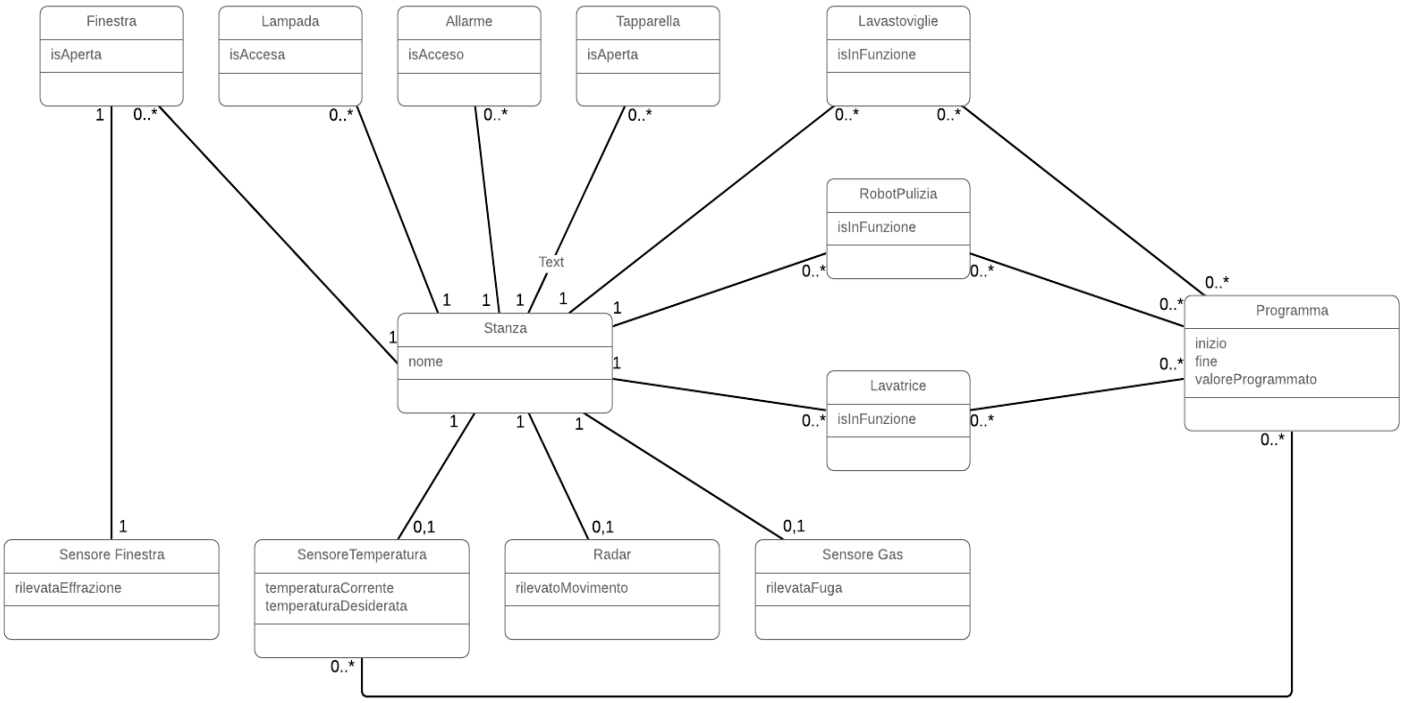


Figura 2: Modello di Dominio

L'idea originale della casa, in fase di analisi, era questa: avere una stanza che contenesse diversi oggetti smart a cui cambiare stato. Inoltre alcuni oggetti potevano essere programmati in diversi giorni a differenti ore.

3 Progettazione

3.1 Architettura Software

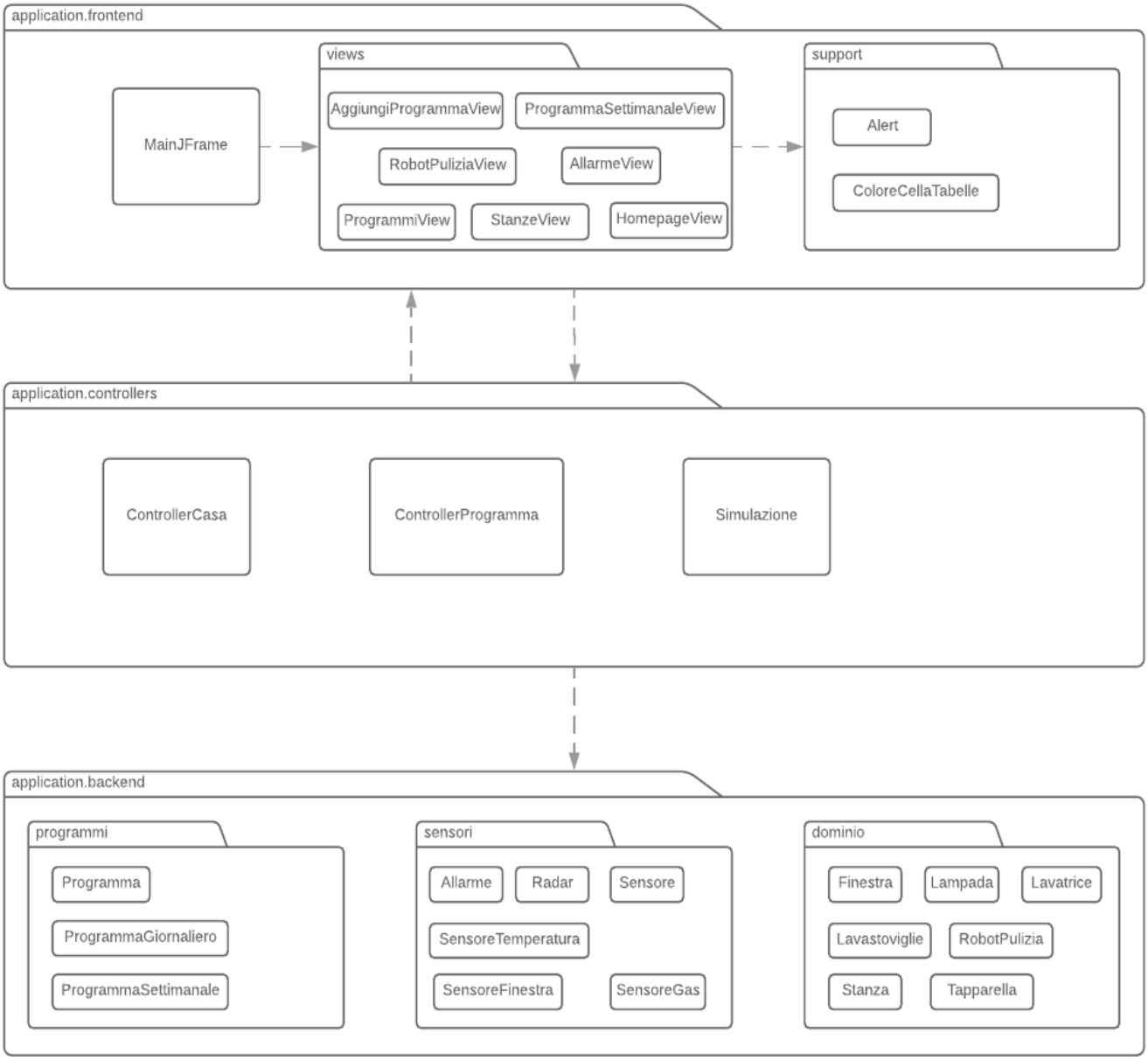
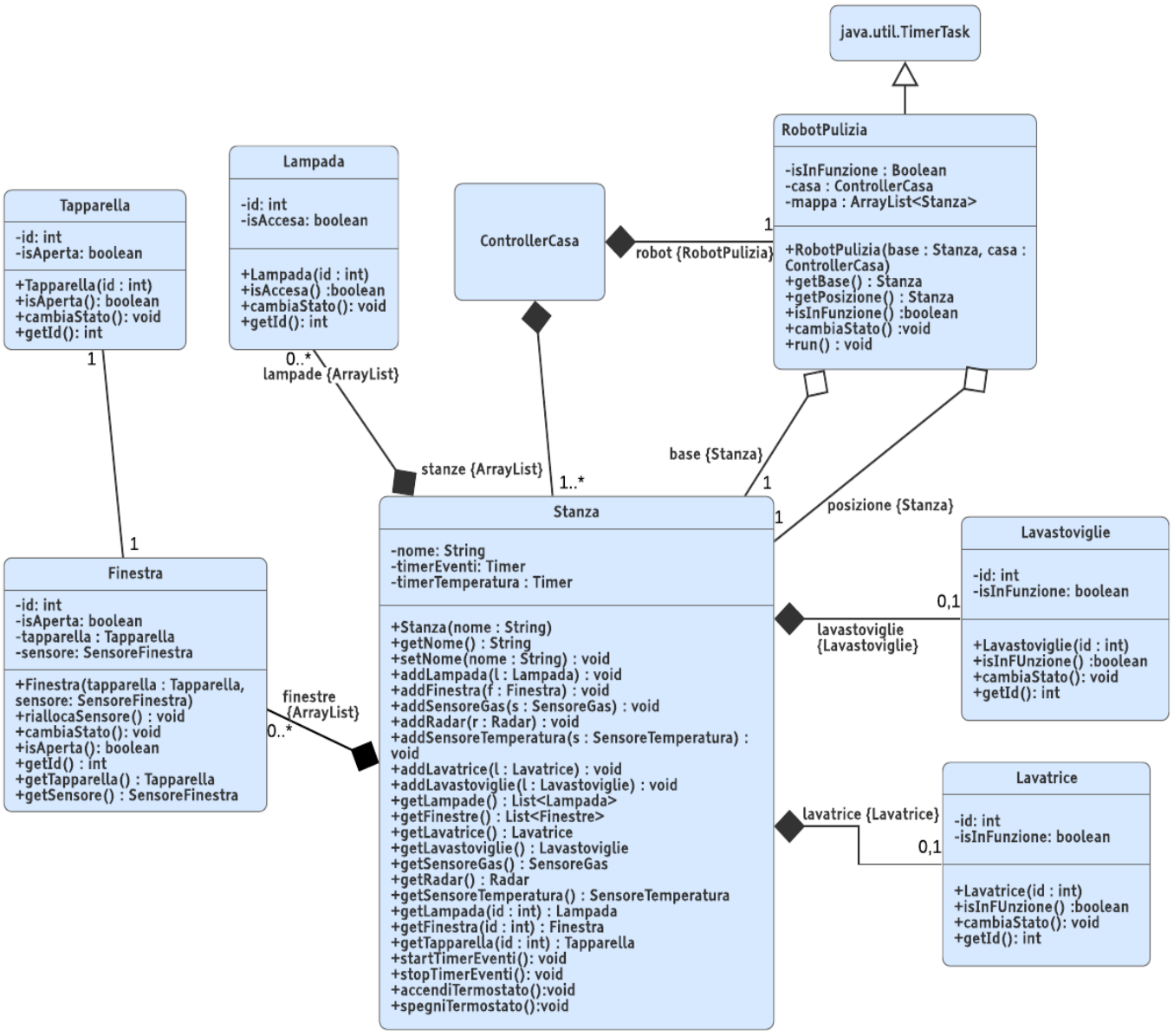


Figura 3: Diagramma dell'architettura software

3.2 Diagramma delle Classi di Progetto

3.2.1 Application.Backend.Dominio



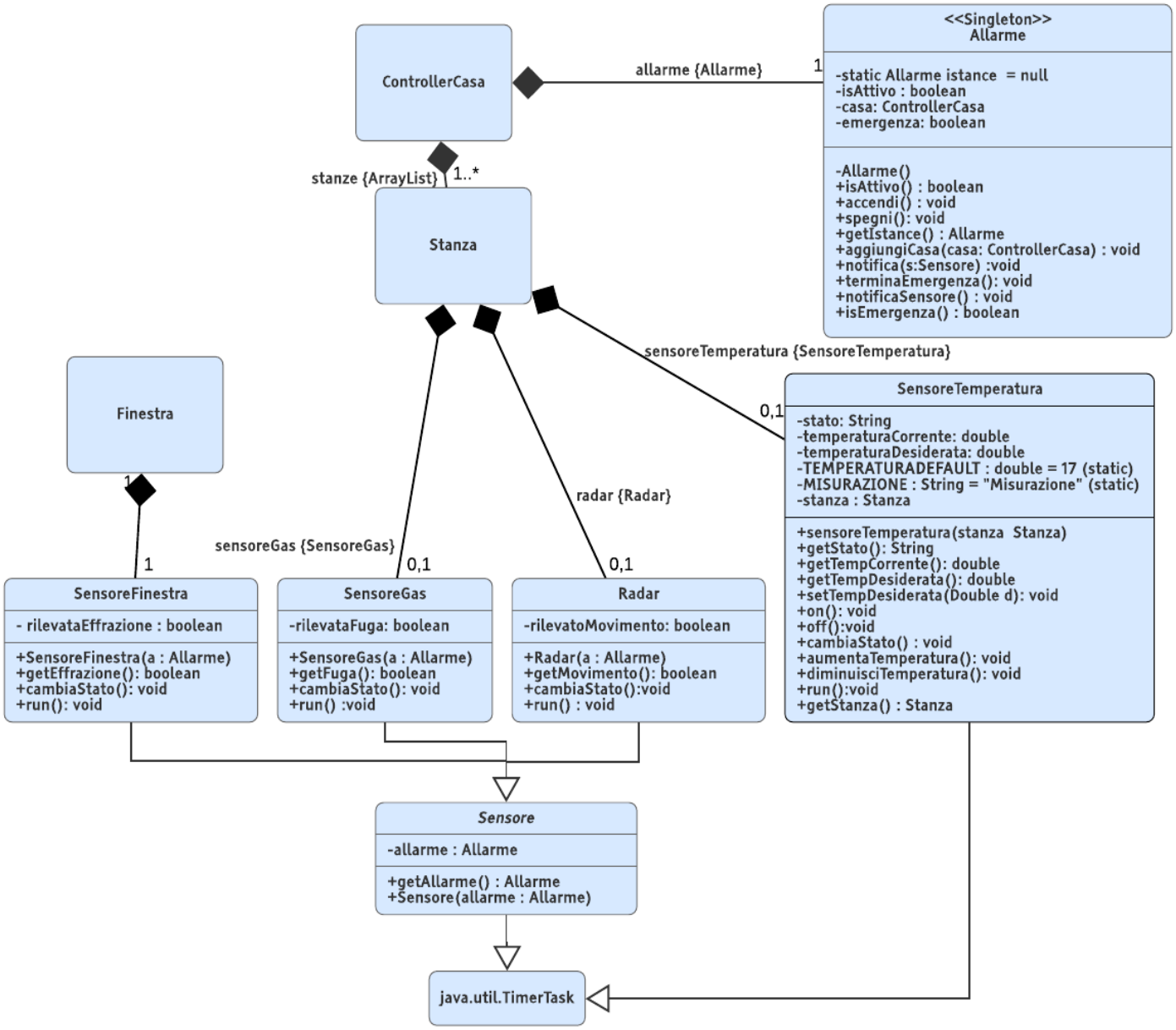
L'oggetto *Stanza* è un oggetto **Facade** in quanto rappresenta il punto di entrata ai singoli elementi; infatti, l'utente non interagisce direttamente con i singoli oggetti, ma interagisce con *Stanza*.

L'utente può compiere le seguenti azioni:

- accendere/spegnere le lampade;
- aprire/chiudere le finestre;

- alzare/abbassare le tapparelle;
- accendere/spegnere la lavatrice;
- accendere/spegnere la lavastoviglie;
- modificare la temperatura della stanza;

3.2.2 Application.Backend.Sensori



Sfruttiamo il concetto di **Observer**, ma ne modifichiamo la implementazione

- *Sensore*: conosce tutti i sensori che si trovano in casa;
- *Allarme*: è la classe che deve essere avvisata sui cambiamenti di *Sensore*, e implementa direttamente le azioni da compiere in risposta al messaggio inviato da *SensoreConcreto*;
- *SensoreConcreto*: contiene i valori che interessano ad *Allarme*. Invia un avviso quando lo stato cambia;

La classe *Allarme* è un **Singleton**; è la centralina. Ne è presente una in ogni casa, infatti ne viene creata una e una sola istanza.

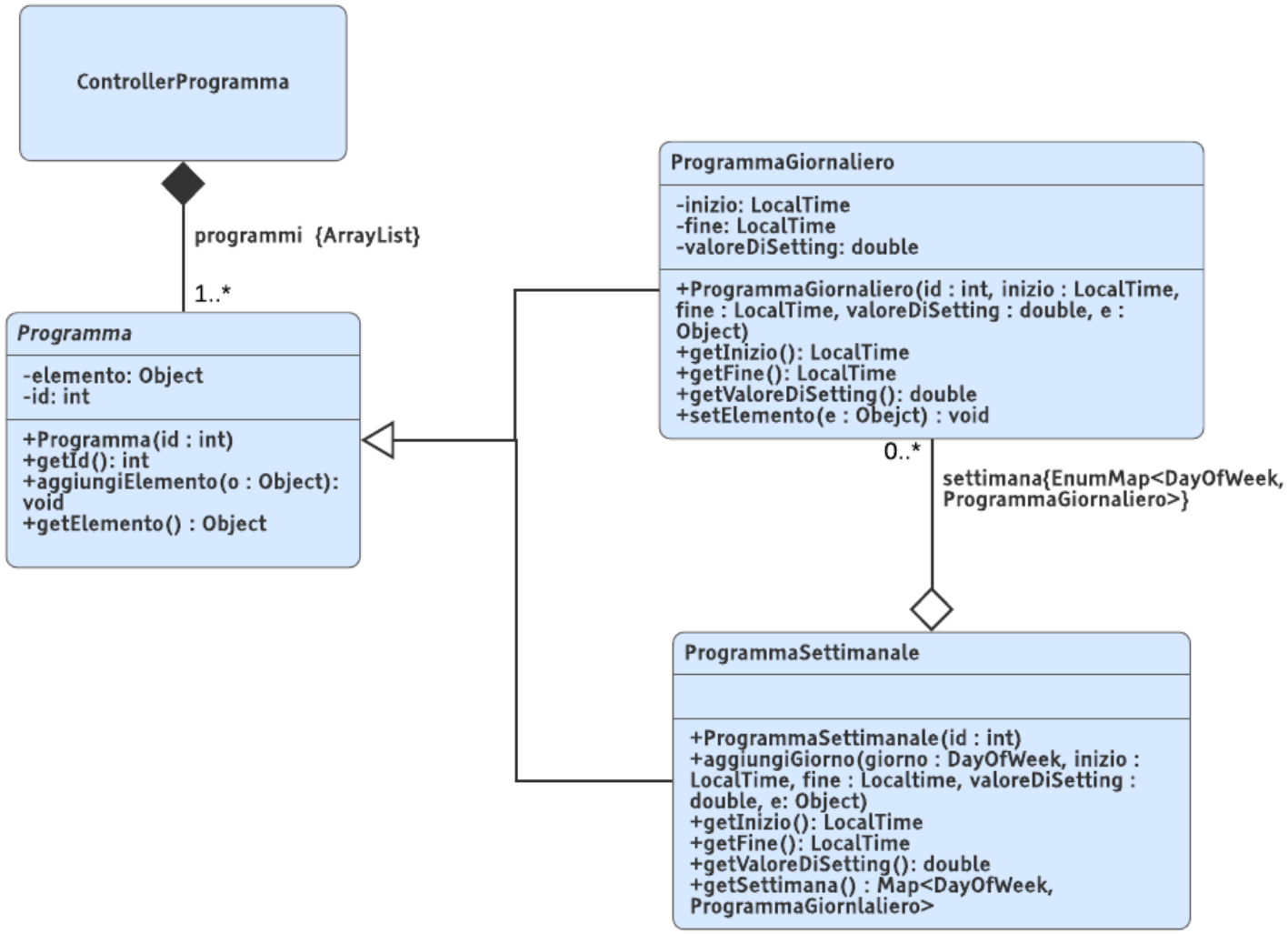
Quando l'allarme viene acceso, i sensori entrano in ascolto nelle singole stanze.
Se vi è un'emergenza, la gestiscono e avvisano l'utente dell'accaduto; esso, per riportare i sensori allo stato iniziale di ascolto, deve terminare la emergenza.

- ***SensoreGas***: rileva se ci sono fughe di gas.
  In caso positivo, si aprono le finestre. Fin quando l'utente non termina l'emergenza, non può chiudere le finestre in quella stanza.
- ***SensoreFinestra***: rileva se ci sono effrazioni.
- ***Radar***: rileva se ci sono movimenti.

Il ***SensoreTemperatura*** non è a contatto diretto con l' allarme, ma comunque rimane in ascolto su una singola stanza: infatti, quando è accesso, controlla la temperatura e decide se regolarla.
La temperatura di default in una stanza è di 17°.
L'utente può decide di impostare una temperatura compresa tra i 8° e i 28° (esclusi gli estremi); in base alla temperatura corrente, il sensore decide di riscaldare o raffreddare la stanza.

3.2.3 Application.Backend.Programma

Grazie ai programmi, l'utente può impostare l'accensione di alcuni elementi, in un determinato giorno e ad una determinata ora.



Usiamo **Strategy** per implementare *Programma* con i seguenti algoritmi, che sono diversi, ma correlati:

- ***ProgrammaGiornaliero***: dura 2 ore e viene eseguito nello stesso giorno in cui viene creato. L'utente non può aggiungere un programma giornaliero che parte ad un orario che è minore di quello in cui esso viene creato.
  (*Esempio: se l'utente, alle 15:00, vuole creare un programma che imposta la partenza della lavatrice alle ore 8:00, il programma non viene aggiunto*).
  Un programma giornaliero, dopo essere terminato, viene cancellato.

- ***ProgrammaSettimanale***: rappresenta la settimana e per ogni giorno vi è un programma giornaliero. L'utente può decidere di programmare solo alcuni giorni.
  A differenza del programma giornaliero, il programma settimanale non viene cancellato automaticamente; inoltre, l'utente può inserire l'orario che vuole, indipendentemente dall'ora e dal giorno corrente.
  (*Esempio: se l'utente, alle 15:00 di Lunedì, vuole creare un programma che imposta la partenza della robot alle ore 8:00 del Lunedì, il programma viene aggiunto. Il lunedi successivo alle ore 8:00 il robot partirà*).

Ogni programma, qualunque esso sia, fa riferimento ad un solo oggetto.

L'utente può creare programmi giornalieri su *Lavatrice* e su *Lavastoviglie*, mentre può creare programmi settimanali su *SensoreTemperatura* e su *RobotPulizia*; inoltre, indipendentemente dal sistema, può cancellare i programmi creati.

Caso di SensoreTemperatura: la scelta attuale dell'utente ha la precedenza sul programma precedentemente creato.
(*Esempio: supponiamo siano le 10:00 di Lunedi e supponiamo esista un programma settimanale che imposti che, un sensore di temperatura, il Lunedì alle 10:00 debba settare la temperatura della stanza a 15°. Se l'utente, ora, dovesse decidere che la temperatura in quella stanza sia di 19°, allora il sensore deve settare la temperatura della stanza a 19°*).

3.2.4 Application.Backend.Controllers

Per evitare che l'utente accedesse direttamente agli oggetti di dominio, utilizziamo il pattern architetturale **Model-View-Controller**, per separare appunto la logica dell'applicazione dai componenti che implementano la grafica.

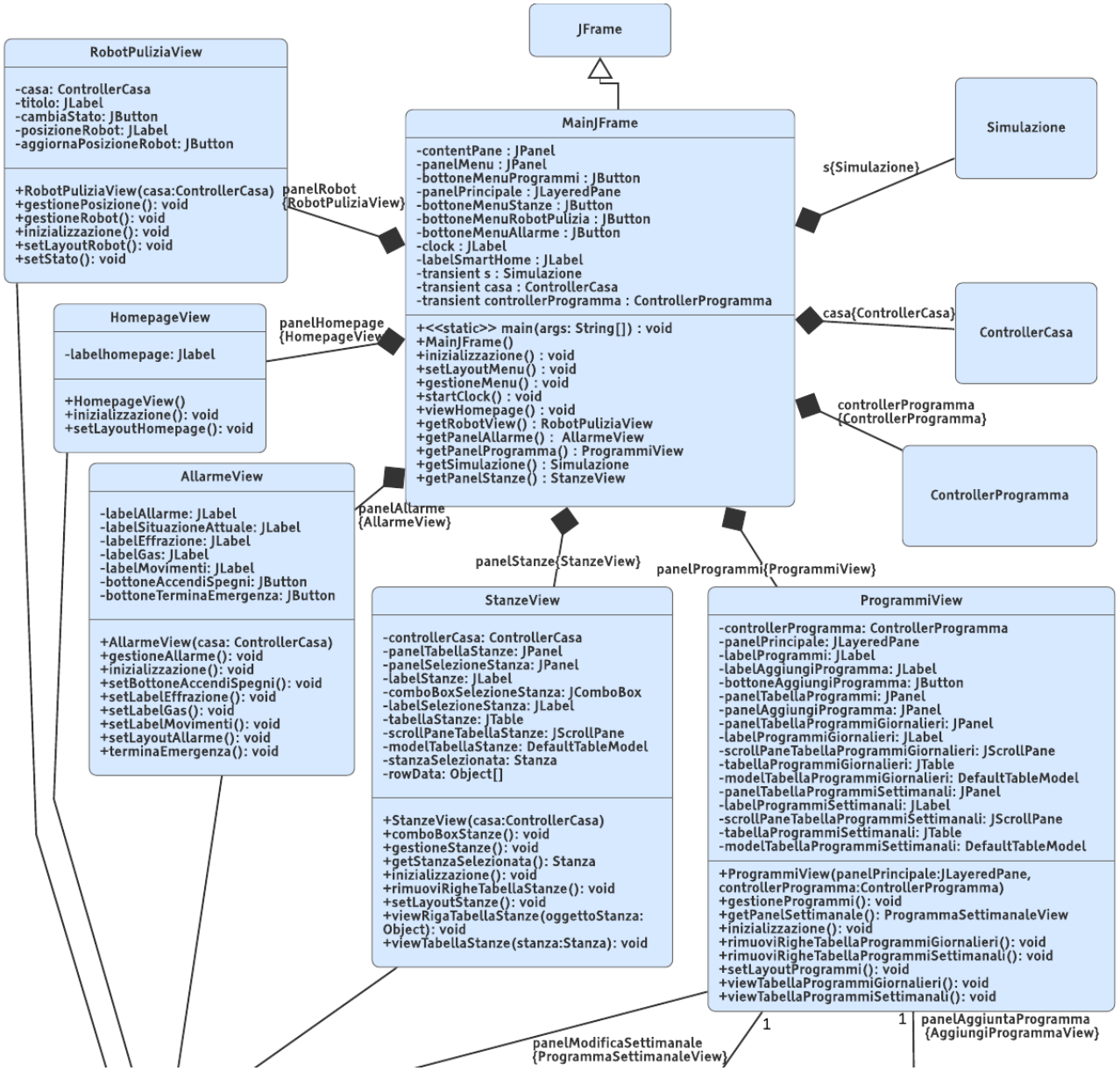
- *Model*: riferimento al package *Application.Backend.Dominio* e *Application.Backend.Programmi*
- *View*: riferimento al package *Application.Frontend*
- *Controller*:
  - ***ControllerCasa***: gestisce tutti gli oggetti che si trovano in una stanza, l'allarme e il robot della pulizia.
  - ***ControllerProgramma***: gestisce i programmi, creati dall'utente.

La classe ***Simulazione*** gestisce lo scorrere del tempo, e costantemente controlla se ci siano dei programmi da far partire.

3.2.5 Application.Backend.Frontend

Abbiamo implementato la grafica utilizzando la libreria grafica di Java chiamata Swing.

La classe *Alert* crea i messaggi di errore/informazione per l'utente.
La classe *ColoreCellaTabella* colora determinate celle delle tabelle dell'applicazione in base a delle caratteristiche (ad esempio lo stato degli elementi).



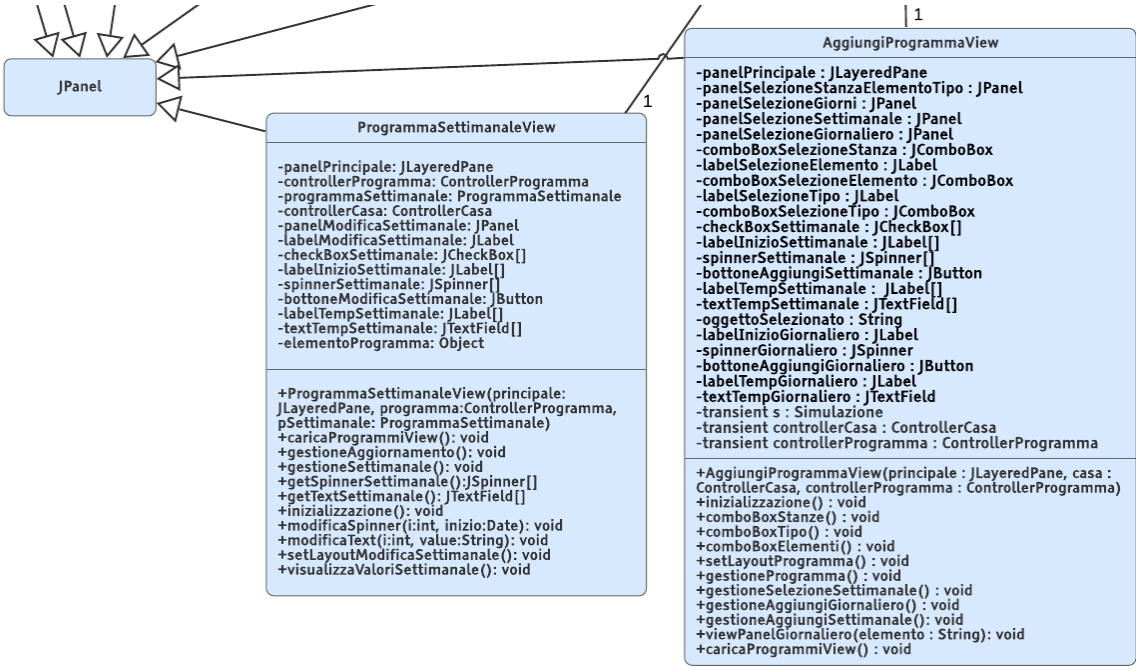


Figura 4: Classi della grafica

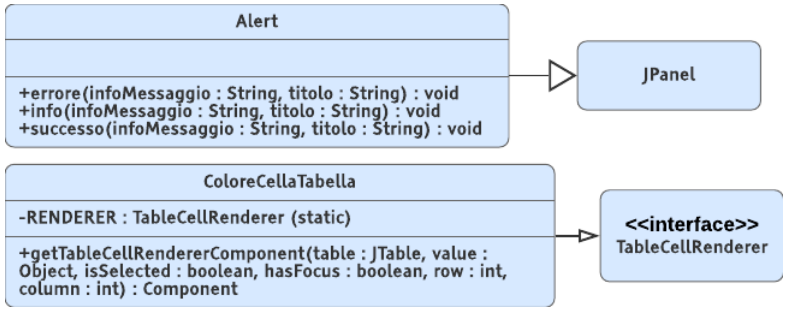


Figura 5: Classi di supporto alla grafica

3.3 Diagramma di Comunicazione

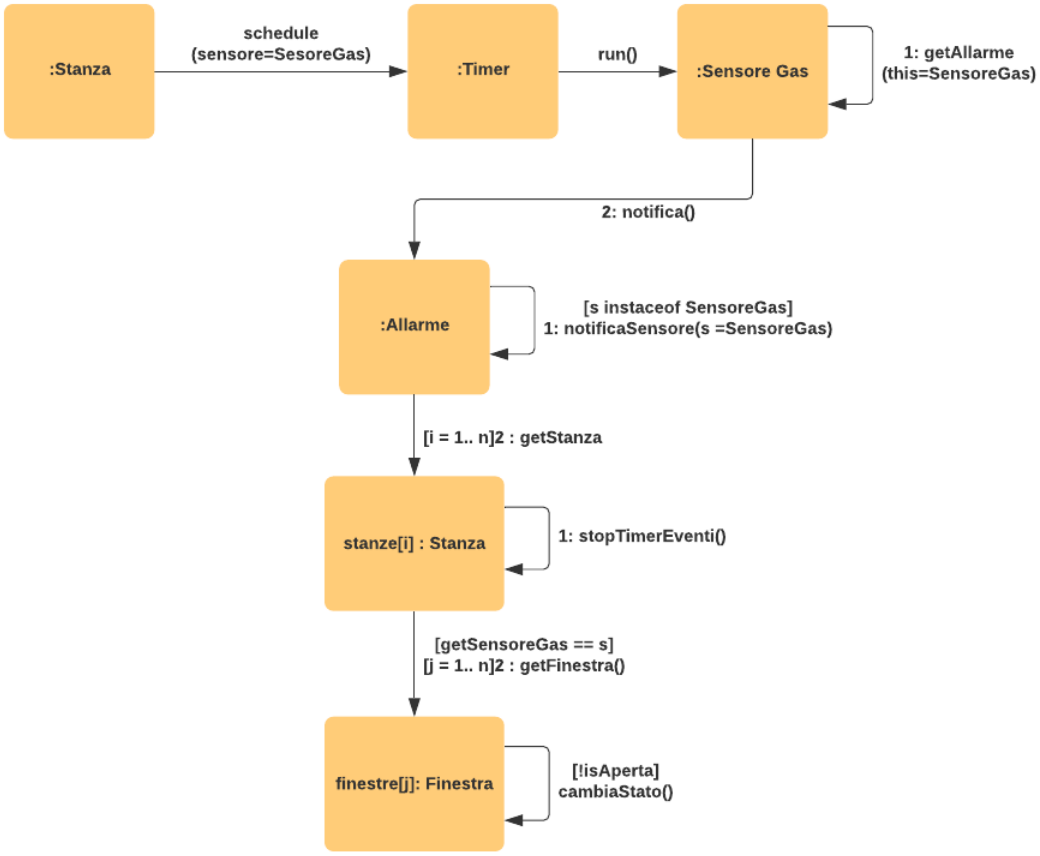


Figura 6: Diagramma di Comunicazione relativo alla generazione dell'evento casuale "Fuga di Gas" e la sua gestione da parte dell'allarme

3.4 Diagramma degli Stati

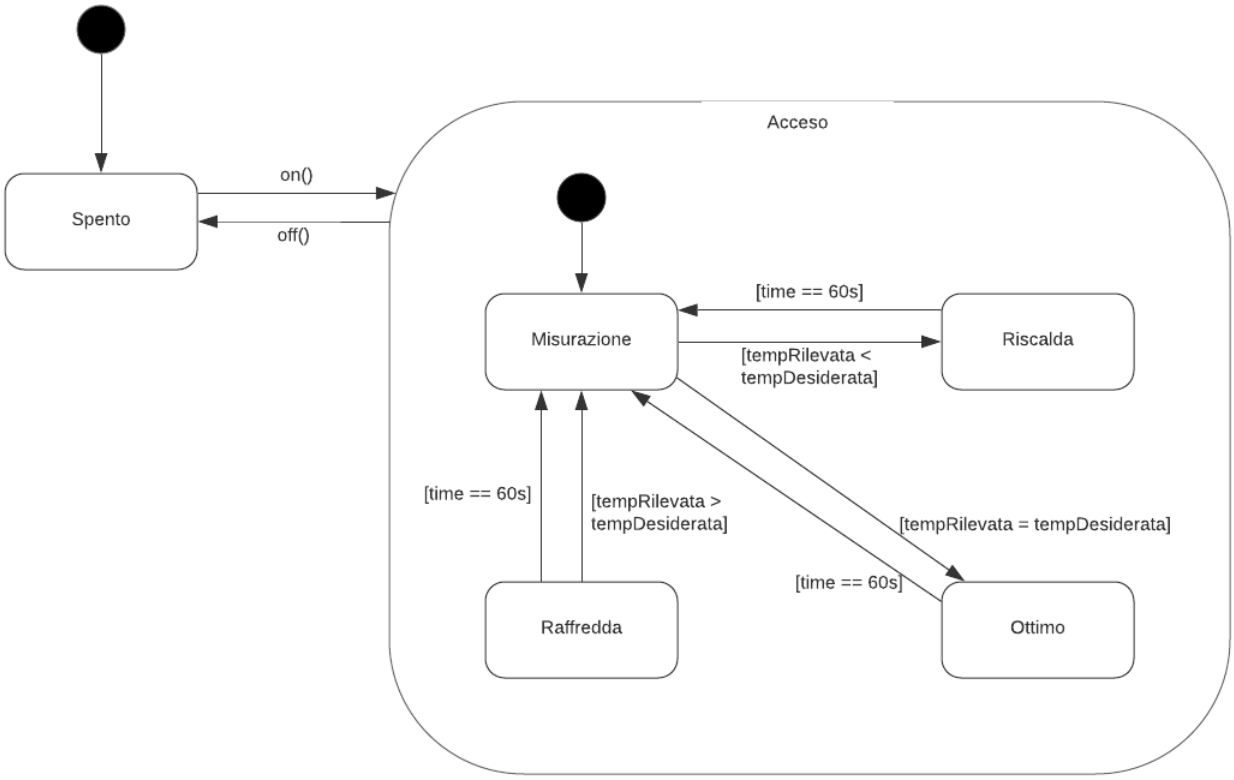


Figura 7: Diagramma degli Stati relativo al SensoreTemperatura

3.5 Diagramma delle Attività

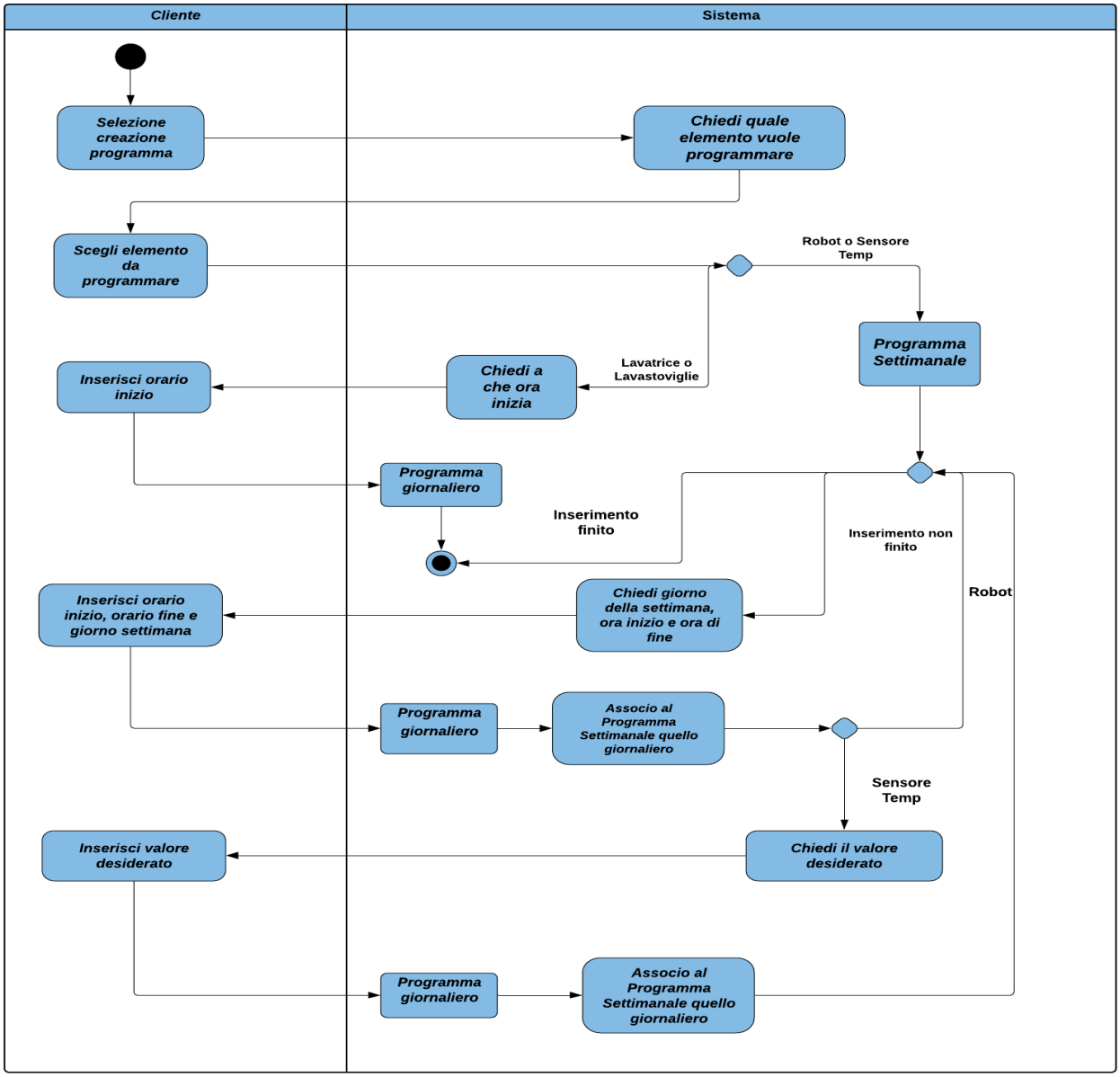


Figura 8: Diagramma delle Attività relativo alla creazione di un Programma

4 Analisi attraverso i Tool

Per l'analisi della qualità del codice abbiamo utilizzato due tool: Understand e SonarQube

4.1 SonarQube

Il nostro codice, analizzato attraverso SonarQube, ha la qualità massima per Bug, Vulnerabilità e Code Smell.

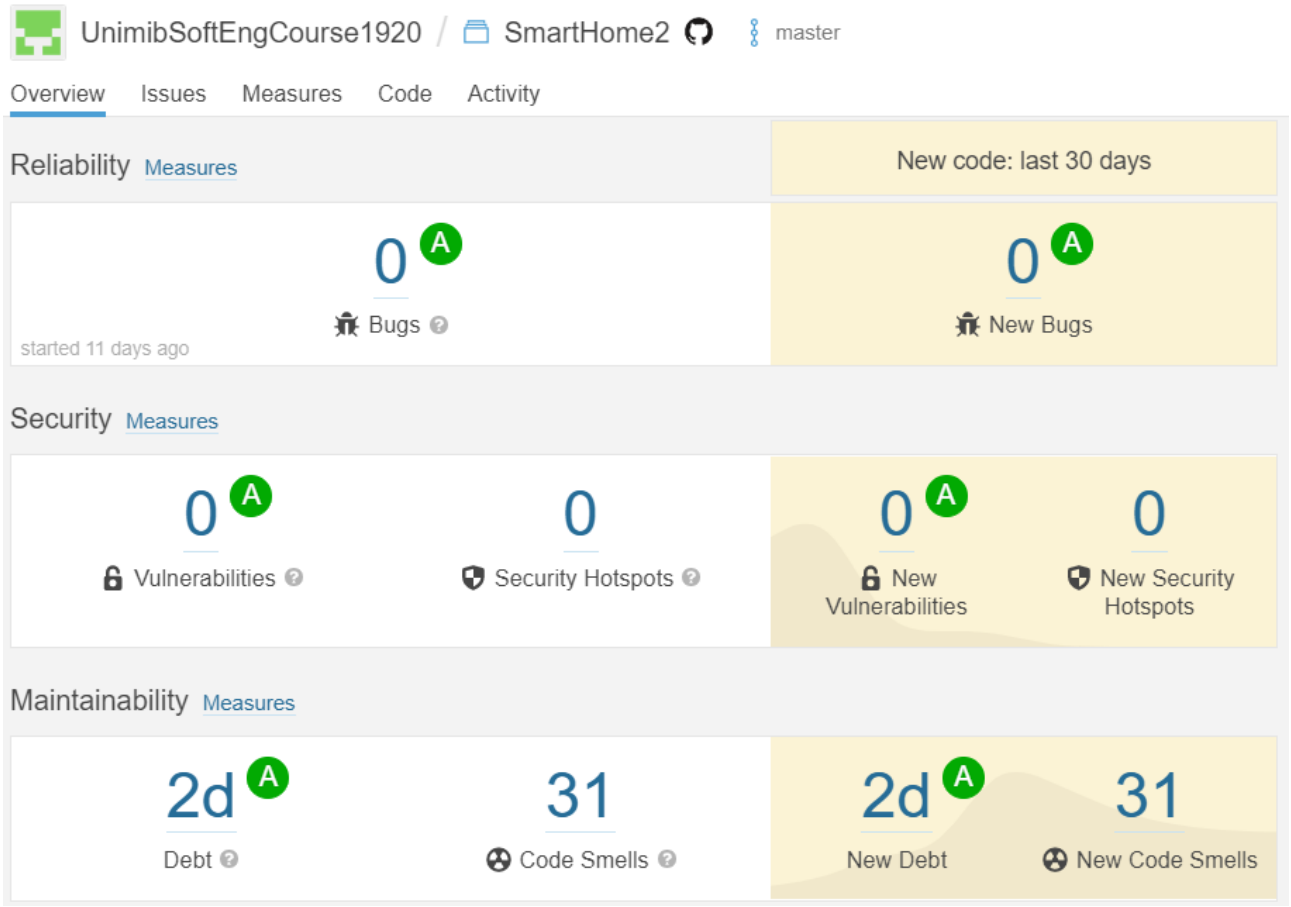


Figura 9: Schermata di SonarCloud al termine dello sviluppo

Come si può notare dalla precedente foto, non tutti i code smell sono stati eliminati, in particolare:

- Refactor di metodi con una complessità cognitiva molto alta, soprattutto nelle classi di gestione della grafica
- Stringhe duplicate per la gestione degli alert degli errori
- Blocchi try/catch innestati, che sarebbero dovuti essere estratti in metodi separati, ma non abbiamo potuto sistemare poichè questi blocchi utilizzavano molte variabili del metodo in cui essi si trovano; se li avessimo spostati, di conseguenza, avremmo dovuto ritornare al metodo molti valori differenti

Inoltre vi è del codice duplicato a causa della complessa gestione della grafica

4.2 Understand

Abbiamo utilizzato Understand per individuare alcuni anti-pattern strutturali del nostro codice

- GlobalHub: Le classi controller e la classe main sono dei GlobalHub poichè contengono molte dipendenze e molti dipendenti, provenienti anche da package differenti. Per definizione le classi controller devono comunicare sia con il dominio che con la grafica e la classe main è il contenitore sia della grafica che dei controller

Inoltre abbiamo creato un report generale sul progetto:

- Linee vuote: 628
- Classi: 66
- Linee di codice: 3.675
- Linee di codice commentato: 40
- Rapporto tra commenti e codice:0,01
- Dichiarazioni: 929
- Esecuzioni: 1.386
- File: 33
- Funzioni:282
- Linee complessive: 4.344

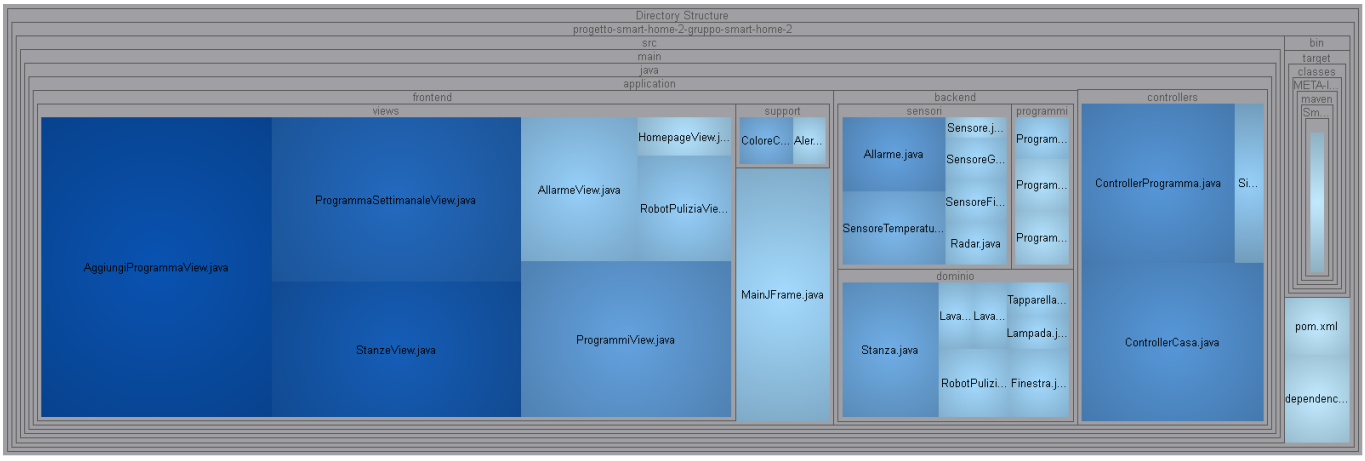


Figura 10: TreeMap che evidenzia la grandezza della classe in termini di linee di codice e la complessità ciclomatica

Come possiamo notare dal TreeMap, le classi con la complessità maggiore sono quelle più grosse in termini di linee di codice. In particolare, la parte grafica richiede una complessità maggiore per la gestione degli input dell'utente

5 Guida all'utilizzo della SmartHome

5.1 Come far eseguire l'applicazione

Il jar eseguibile dell'applicazione con il nome di "EseguibileSmartHome2.jar" si trova nella cartella root del progetto.
Aprire il prompt dei comandi: dopo essere arrivati dentro la cartella di root del progetto, eseguire il seguente comando:

java -jar EseguibileSmartHome2.jar

e l'applicazione partirà, mostrando la seguente schermata:

Figura 11: Schermata Iniziale

Si può visualizzare il clock, in alto a sinistra, che scorre in tempo reale.
L'utente, dal menù, può scegliere di visualizzare una delle 4 schermate:

- Programmi;
- Stanze;
- RobotPulizia;
- Allarme;

Quando vuole può premere su "SMART HOME" e tornare alla schermata iniziale.

5.2 Sezione Programmi

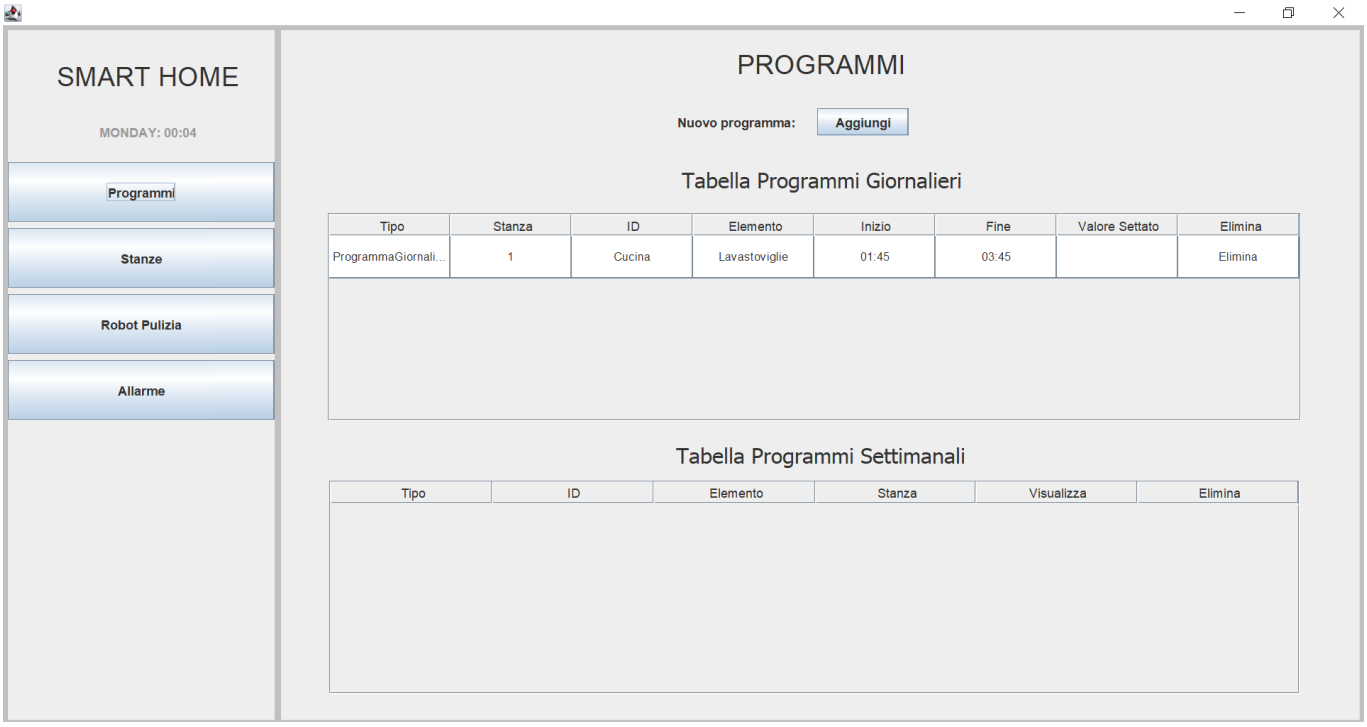


Figura 12: Schermata Programmi

Da qui, l'utente può visualizzare, mediante delle tabelle, i programma creati. (E' già stato inserito un programma giornaliero di prova).

Premendo il pulsante "Aggiungi" si aprirà la schermata relativa all'aggiunta di un programma (settimanale o giornaliero).

5.2.1 Aggiunta di un Programma Giornaliero

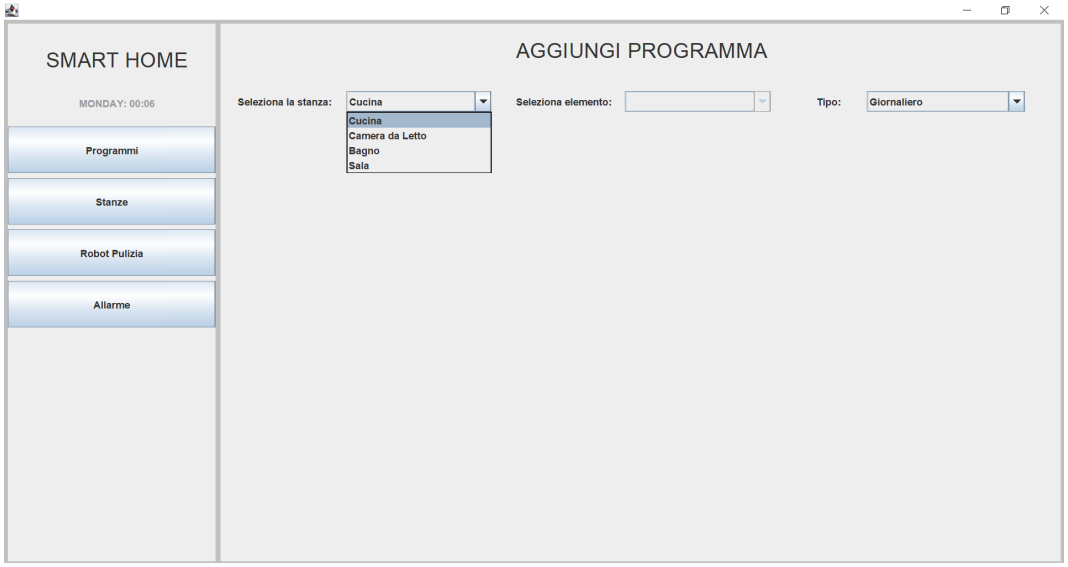


Figura 13: Scelta Stanza

L'utente deve scegliere su quale oggetto creare il programma; seleziona prima la stanza, e poi l'oggetto che si trova in quella stanza.

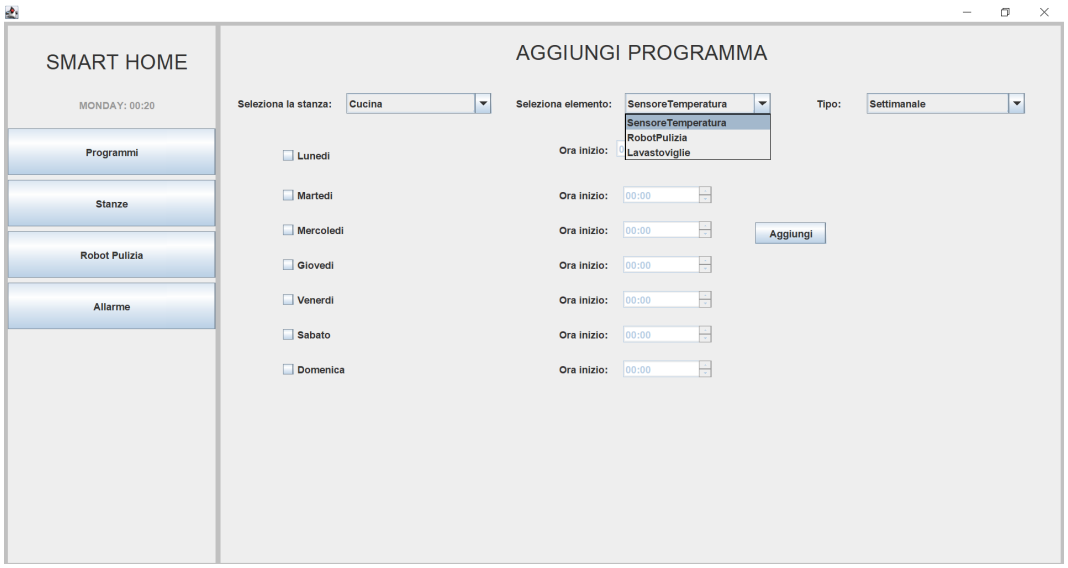


Figura 14: Scelta elemento

In base all'elemento selezionato, il programma da creare o è giornaliero o è settimanale.
(Si ricorda che si possono creare programmi giornalieri solo su lavatrice e lavastoviglie, mentre programmi settimanali su robotPulizia e sul sensore della temperatura).

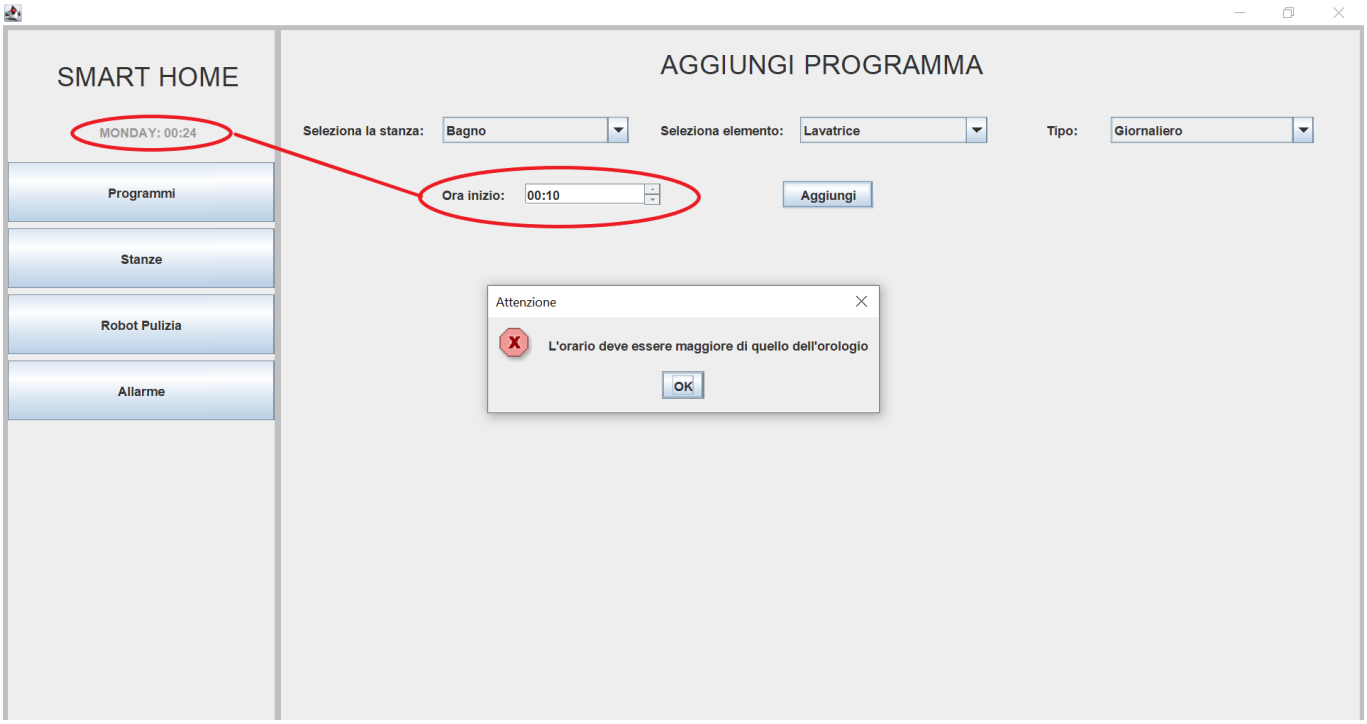


Figura 15: Aggiunta Programma Giornaliero

L'utente deve stare attento nell'inserire l'orario di inizio; infatti, siccome il programma giornaliero fa riferimento al giorno corrente, e quindi l'elemento scelto dovrà essera azionato nel giorno corrente, non si può inserire un orario di inizio che è minore di quello corrente.
Se l'utente dovesse inserire, per sbaglio, un orario minore, il sistema lo informerà dell'errore.
Altrimenti, il programma viene aggiunto e l'utente viene reinderizzato alla pagina della visualizzazione dei programmi.

NB: un programma giornaliero creato non può essere modificato! Per essere modificato, deve essere cancellato e poi ricreato.

5.2.2 Aggiunta di un Programma Settimanale

Dopo aver scelto la stanza e l'elemento, come per il programma giornaliero, comparirà questa schermata:

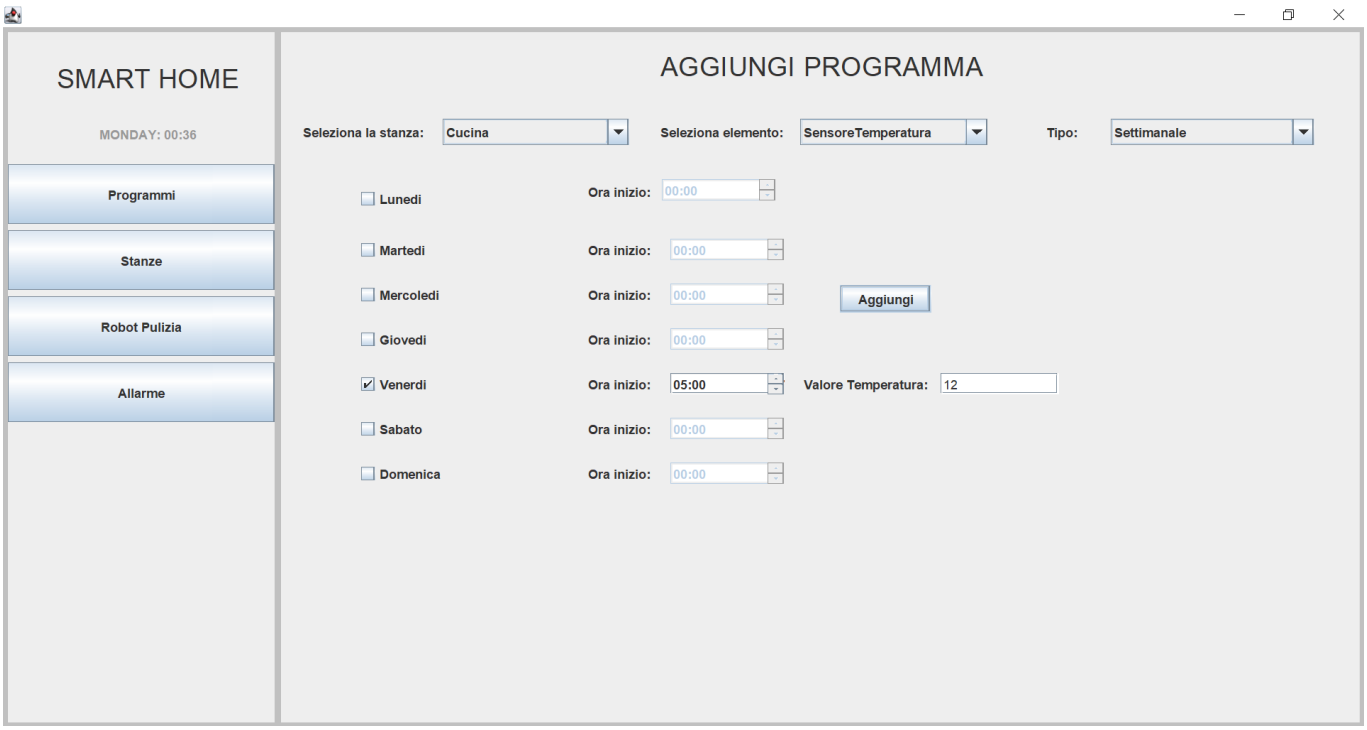


Figura 16: Aggiunta Programma Settimanale

A differenza del programma giornaliero, qui l'utente non ha vincoli; può inserire l'orario che più preferisce.
L'utente non è obbligato ad impostare, in questo caso, la temperatura desiderata per tutti i giorni, ma può impostarla solo per alcuni, come nel caso in figura.
L'utente pu'o decide di impostare una temperatura compresa tra i 8° e i 28° (esclusi gli estremi).
NB: l'utente deve ricordarsi, però, di **"attivare"** il giorno, altrimenti il giorno non viene considerato.
Premendo sul pulsante "Aggiungi", il programma settimanale viene aggiunto e l'utente viene reinderizzato alla pagina della visualizzazione dei programmi.

5.2.3 Modifica di un Programma Settimanale

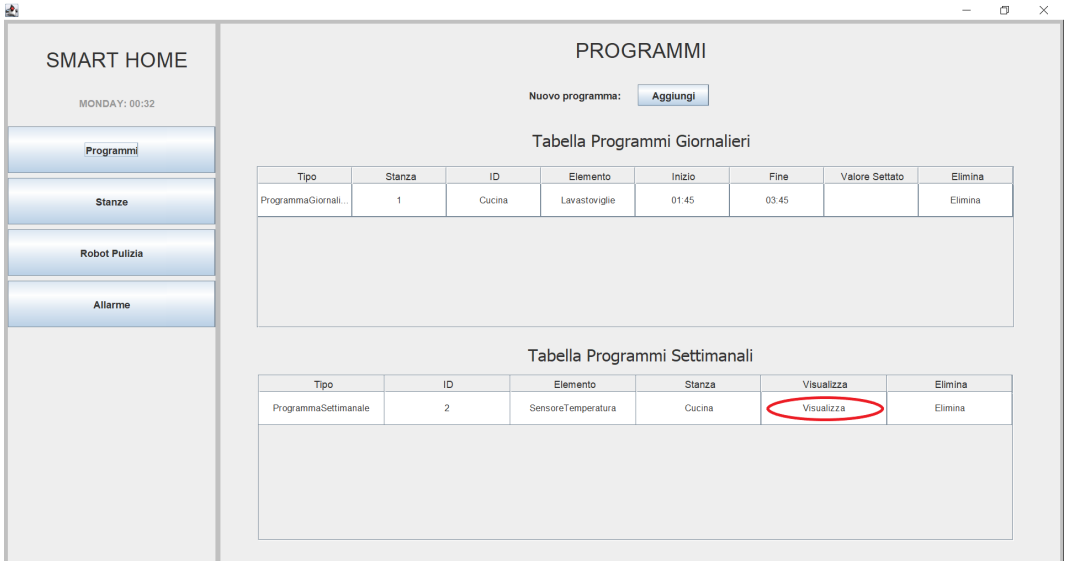


Figura 17: Visualizza Programma Settimanale

Dalla schermata della visualizzazione dei programmi, premere su "Visualizza" per visualizzare il programma settimanale.

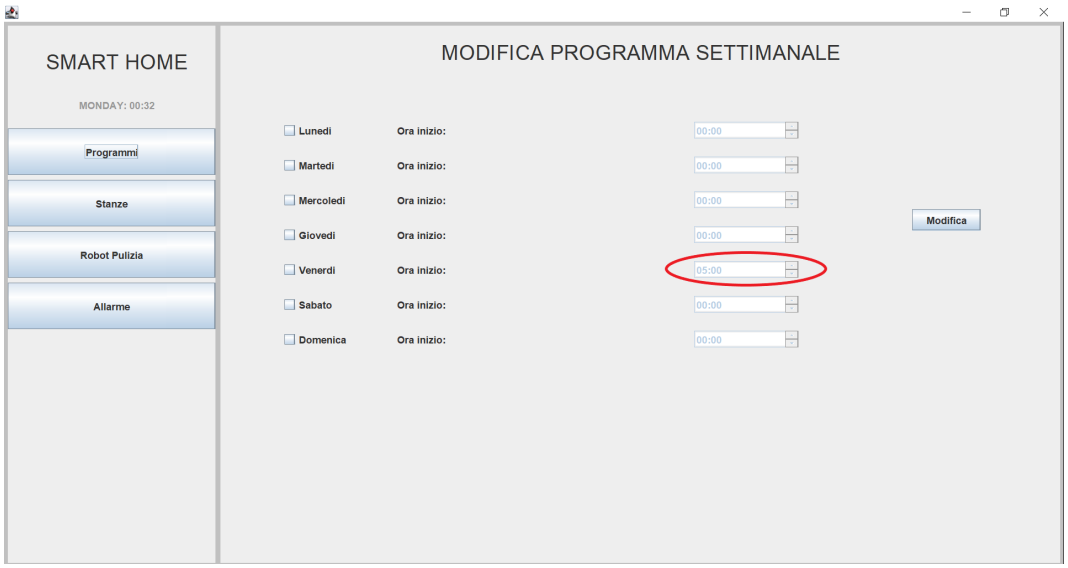


Figura 18: Modifica Programma Settimanale

L'utente, a questo punto, se vuole **solo visualizzare** il programma settimanale precedentemente inserito, può notare che vengono visualizzati gli orari scritti in fase di aggiunta.
Non deve premere il tasto "Modifica".

Altrimenti, se vuole **modificare** il programma settimanale precedentemente inserito, può farlo; funziona come per l'inserimento dello stesso.
Se vuole che in un determinato giorno l'elemento parta a una determinata ora, deve **"attivare"** il giorno, e inserire l'orario a scelta.
Premendo sul pulsante "Modifica", il programma settimanale viene modificato e l'utente viene reinderizzato alla pagina della visualizzazione dei programmi.

5.2.4 Eliminazione di un Programma

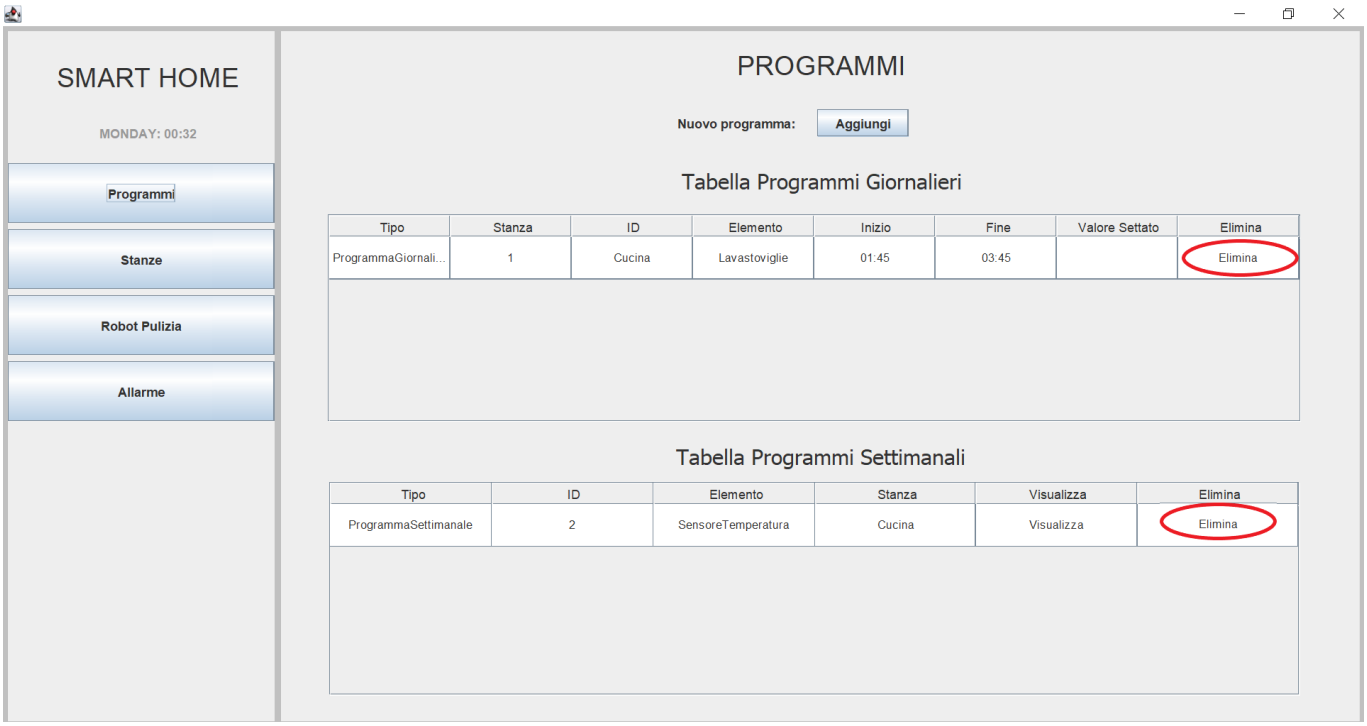


Figura 19: Eliminazione Programma

Indipendentemente che il programma sia giornaliero o settimanale, se l'utente vuole eliminarlo deve premere su "Elimina", e il programma viene eliminato.

5.2.5 Attivazione di un Programma

Il sistema controlla constantemente se in un determinato momento il programma deve partire.
L'utente può verificare la effittiva partenza del programma premendo su "Stanze", scegliere la stanza dove l'elemento è situato e vedere se effettivamente entra in funzione.
NB: il sensoreTemperatura deve essere acceso per far sì che il programma parta.

Si ricorda che una volta chiusa l'applicazione, i programmi salvati andranno persi.

5.3 Sezione Stanze

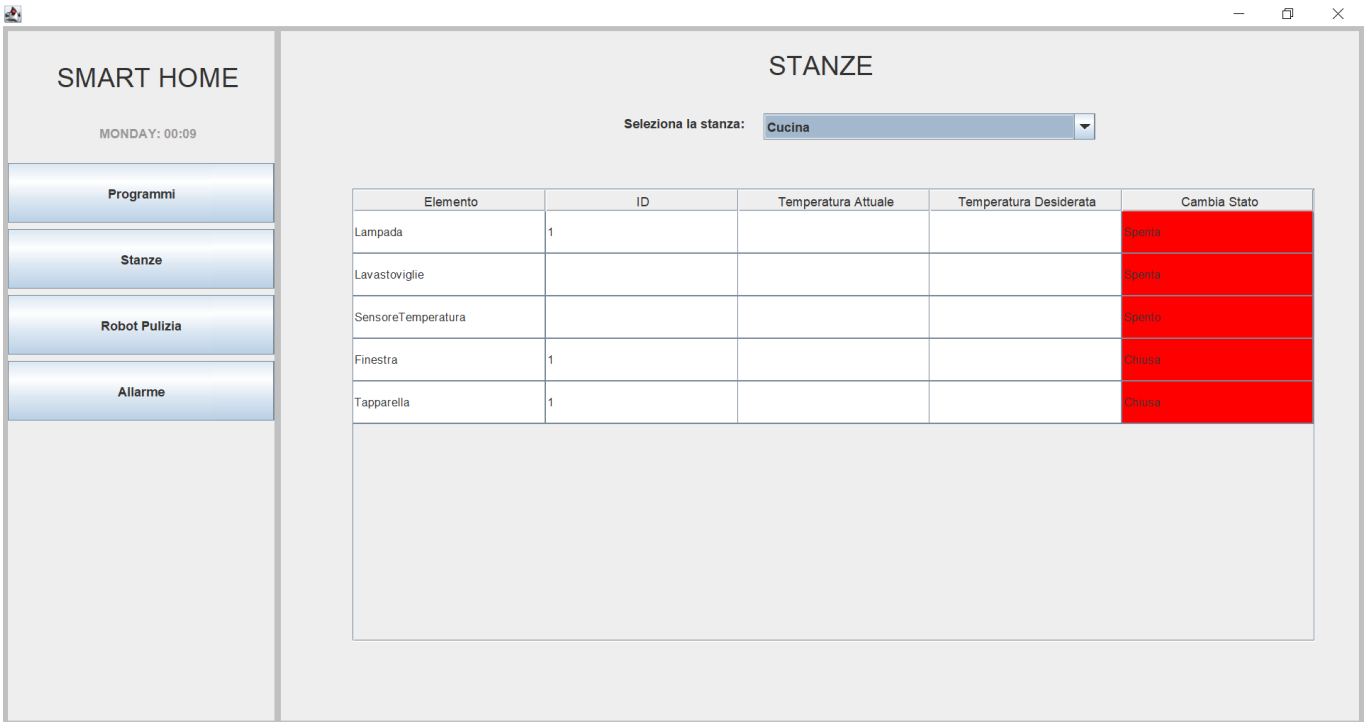


Figura 20: Schermata Stanze

Da qui, l'utente può visualizzare, mediante la tabella, gli elementi della stanza.
Premendo su una cella, in corrispondenza della colonna "Cambia Stato", l'elemento che si trova su quella riga cambia stato. Quindi l'utente può:

- accendere/spegnere le lampade;
- aprire/chiudere le finestre;
- alzare/abbassare le tapparelle;
- accendere/spegnere la lavatrice;
- accendere/spegnere la lavastoviglie;
- accendere/spegnere il sensoreTemperatura;

Nel caso particolare di sensoreTemperatura, quando esso è acceso, l'utente può automaticamente inserire la temperatura che vuole

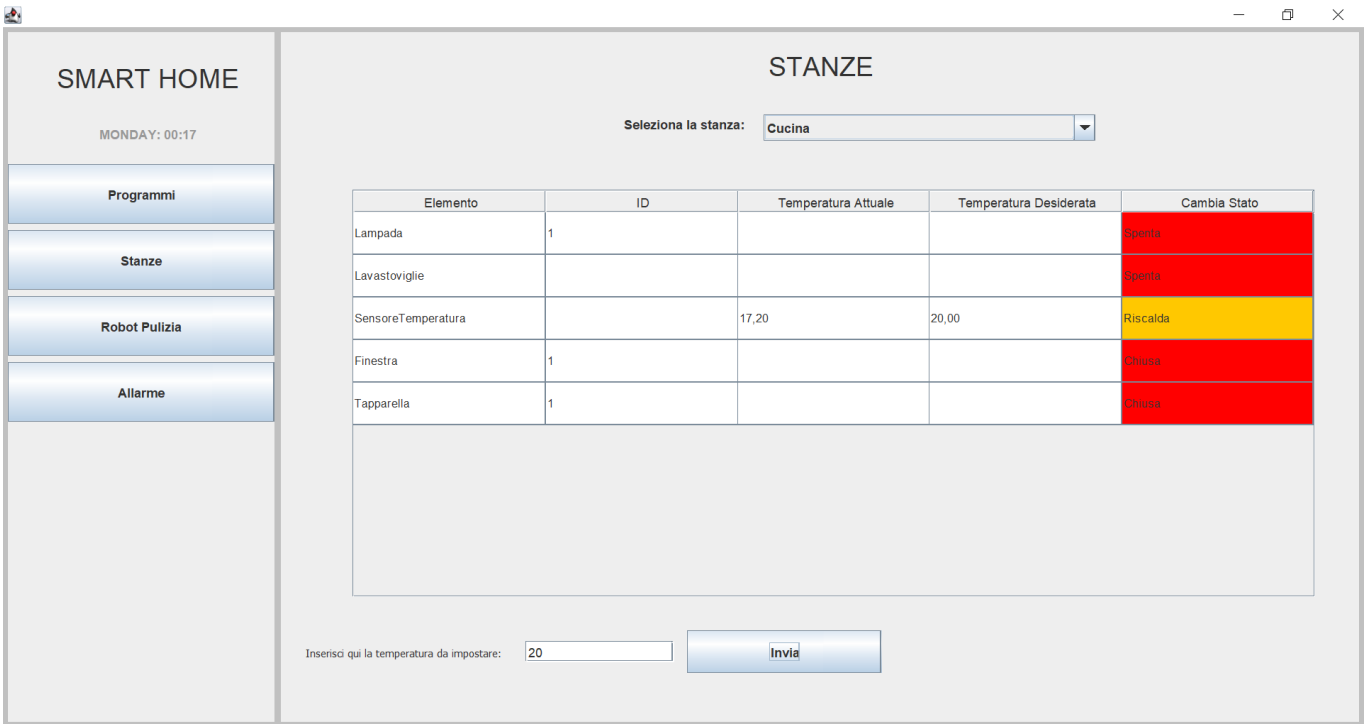


Figura 21: Modifica Temperatura

NB: se c'è un programma impostato su quel sensore temperatura in cucina, ma l'utente in quel momento decide di cambiare la temperatura, il sistema da la precedenza alla scelta attuale dell'utente, quindi il programma viene ignorato.

5.4 Sezione Robot

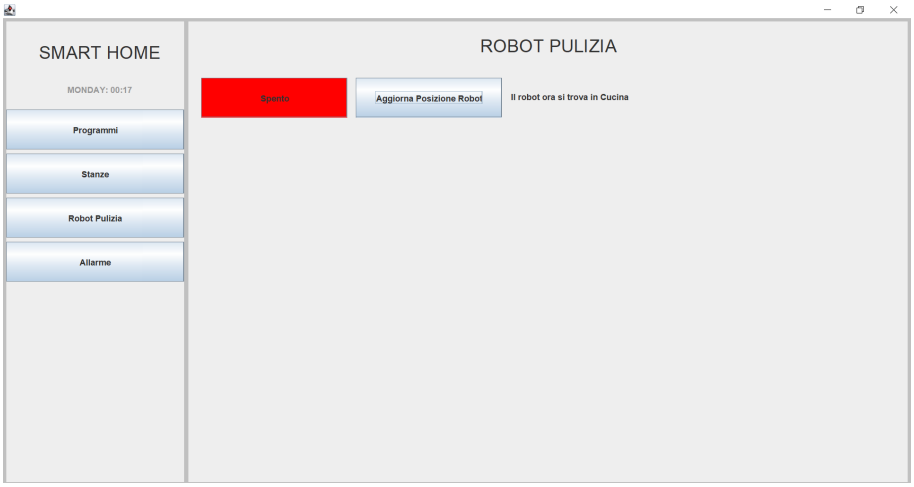


Figura 22: Robot Spento

Il robot si trova momentaneamente spento. La base del robot è in cucina.

L'utente può azionare il robot in qualunque momento premendo su "Spento", così il robot si accende.
Una volta accesso, il robot parte; impiega 2 ore a pulire la casa, e ogni mezz'ora (tempo della simulazione) cambia stanza. Si può tenere controllata la posizione del robot premendo su "Aggiorna Posizione Robot".

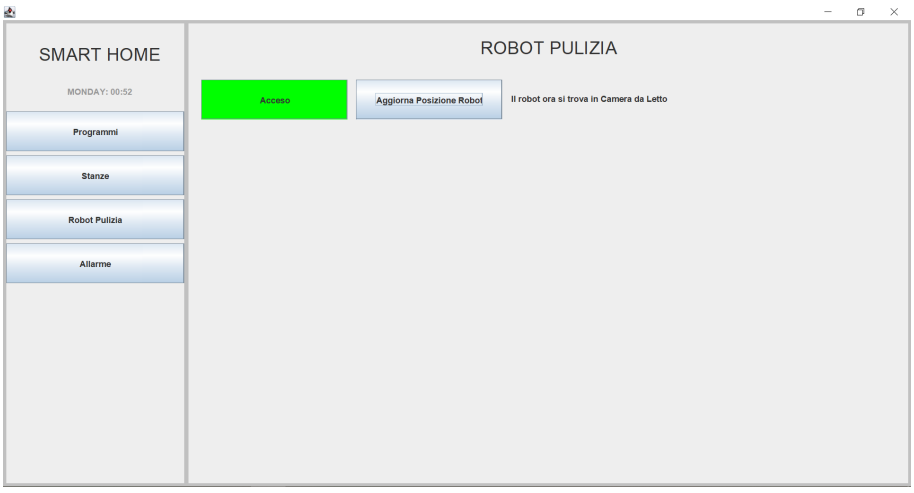


Figura 23: Robot Acceso

Analogamente, l'utente può spegnere il robot in qualunque momento premendo su "Accesso".
Una volta spento il robot torna alla base in cucina.

Se c'è un programma attivo sul robot, il robot parte in automatico.

5.5 Sezione Allarme

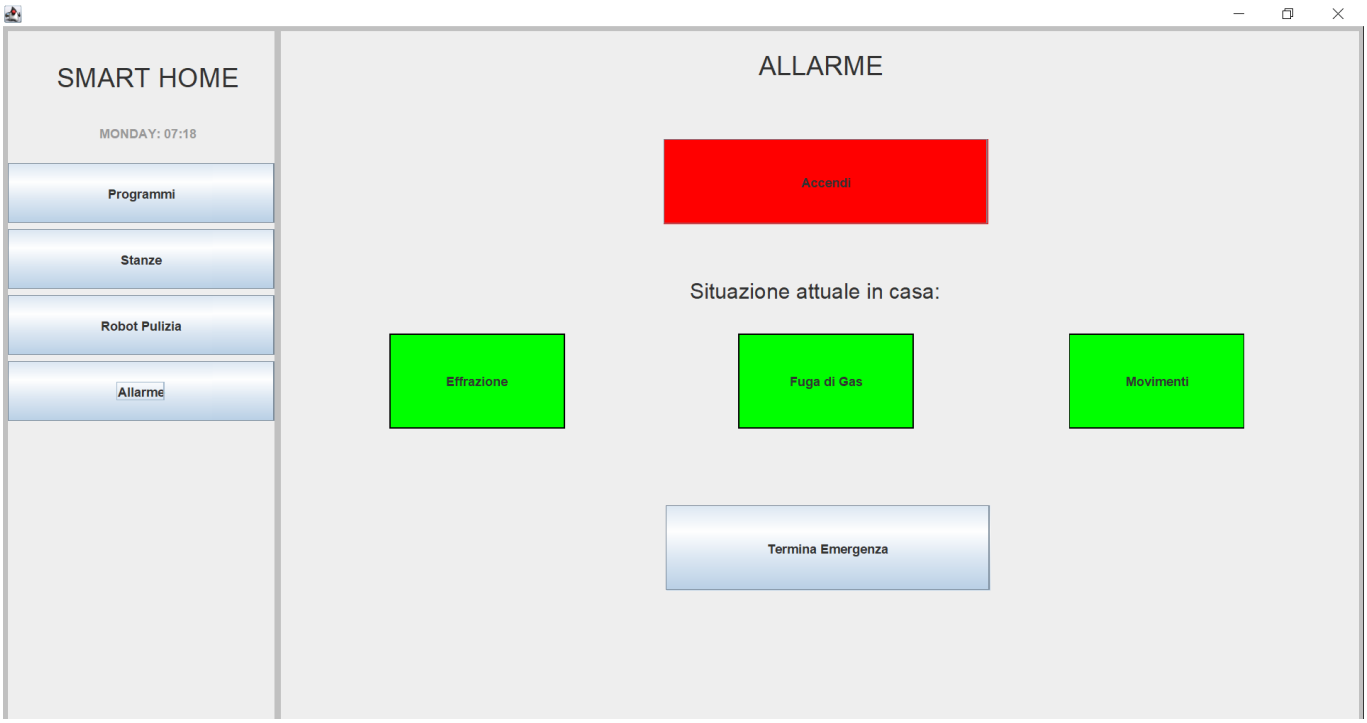


Figura 24: Sezione Allarme

L'utente deve accendere l'allarme, premendo su "Accendi", per far sì che vengano rilevate le fughe di gas, le effrazioni e i movimenti.

Il sistema, con una certa probabilità, simula gli eventi citati prima.
Quando si verifica un evento, il riquadro, relativo all'evento appena capitato, si colora di rosso e l'utente viene informato dell'accaduto, e l'allarme gestisce la emergenza.
L'utente deve premere il bottone "Termina Emergenza" per riportare la casa allo stato iniziale; tutti i riquadri ritorneranno verdi.

Nel caso particolare di una fuga di gas, l'allarme automaticamente apre le finestre nella stanza in cui il sensore ha rilevato la fuga. L'utente non può chiudere quelle finestre fin quando non "termina" l'emergenza.

6 Conclusioni

6.1 Considerazioni Finali

Inizialmente avevamo pianificato anche la progettazione e l'implementazione di altre features, tra cui:

- la scelta della durata di un programma, da parte dell'utente;
- la aggiunta/eliminazione di elementi della casa e delle stanze;
- la gestione del sensoreTemperatura più realistica, ovvero creare un sensore in grado di rilevare autonomamente la temperatura in casa ed esterna e comportarsi di conseguenza, creando automaticamente dei programmi;
- un'interfaccia grafica più "bella"; infatti, avevamo iniziato a progettare la parte grafica con il tool grafico JavaFX. Purtroppo, abbiamo scoperto che lo "studio" della libreria (siccome entrambe non la conoscevamo) e la gestione complessa di essa, dal punto di vista dei thread, richiedeva troppo tempo per riuscire a portare il progetto a termine; di conseguenza abbiamo dovuto rimediare, affidandoci a una libreria a noi conosciuta, e già utilizzata in passato, ovvero Swing.

Svolgendo il progetto in due e, con l'arrivo imminente della consegna, il risultato è stato un piccolo applicativo che funziona nonostante non abbia le richieste scritte sopra.